

**CSCI Integration Test (CIT) Procedures**

**Data Bank Shuttle Automated Function Executive (DBSAFE)**

**Checkout and Launch Control Systems (CLCS)**

**84K06581-001-02**

Approval:

\_\_\_\_\_  
NASA, Quality Assurance                      Date

\_\_\_\_\_  
Test Conductor (T/C)                      Date

\_\_\_\_\_  
System Test                      Date

\_\_\_\_\_  
CSCI Lead                      Date

\_\_\_\_\_

NOTE: See "Supporting Document Note" on following page

**PREPARED BY:** Joe Regan

---

---

---

---

---

---

---

---

**Supporting Document Note:** Acronyms and definitions of many common CLCS terms may be found in the following documents: CLCS Acronyms 84K00240 and CLCS Project Glossary 84K00250.

**REVISION HISTORY**

<b>REV</b>	<b>DESCRIPTION</b>	<b>DATE</b>
Basic	CLCS Redstone Delivery CSCI Integration Test Procedures for DBSAFE - Document, 84K06515	10/21/97
A	CLCS Thor Delivery CSCI Integration Test Procedures for DBSAFE - Document, 84K06581-000-02	01/27/98
Basic	CLCS Thor3 Delivery CSCI Integration Test Procedures for DBSAFE – Document, 84K06581-001-02	06/30/98

LIST OF EFFECTIVE PAGES				
Dates of issue of change pages are:				
Page No.	A or D*	Issue or Change No.	CR No.	Effective Date**

## Table of Contents

<b>1.</b>	<b>SCOPE.....</b>	<b>1-1</b>
1.1	IDENTIFICATION .....	1-1
1.2	PURPOSE .....	1-1
1.3	CSCI OVERVIEW.....	1-1
1.4	HARDWARE AND SOFTWARE CONFIGURATIONS .....	1-2
1.5	DOCUMENT ORGANIZATION .....	1-3
<b>2.</b>	<b>APPLICABLE DOCUMENTATION.....</b>	<b>2-1</b>
2.1	PARENT DOCUMENTS.....	2-1
2.2	APPLICABLE DOCUMENTS .....	2-1
2.3	REFERENCE DOCUMENTS .....	2-2
<b>3.</b>	<b>TEST CASE DESCRIPTION .....</b>	<b>3-3</b>
3.1	TEST CASE 3.1 - TEST LOGICAL DESTINATION SYSTEM FDS.....	3-3
3.1.1	<i>Test Description</i> .....	3-3
3.1.1.1	Detailed Description.....	3-3
3.1.1.2	Resource Requirements .....	3-3
3.1.1.2.1	Test Personnel.....	3-3
3.1.1.2.2	Hardware .....	3-3
3.1.1.2.3	Software.....	3-3
3.1.1.2.4	Data .....	3-3
3.1.1.3	Requirements Summary.....	3-4
3.1.2	<i>Pass/Fail Criteria</i> .....	3-4
3.1.3	<i>Procedure</i> .....	3-4
3.2	TEST CASE 3.2 - TEST PSEUDO (STRING) FDS.....	3-8
3.2.1	<i>Test Description</i> .....	3-8
3.2.1.1	Detailed Description.....	3-8
3.2.1.2	Resource Requirements .....	3-8
3.2.1.2.1	Test Personnel.....	3-8
3.2.1.2.2	Hardware .....	3-8
3.2.1.2.3	Software.....	3-8
3.2.1.2.4	Data .....	3-8
3.2.1.3	Requirements Summary.....	3-9
3.2.2	<i>Pass/Fail Criteria</i> .....	3-9
3.2.3	<i>Procedure</i> .....	3-9
3.3	TEST CASE 3.3 - TEST FD NAME AND NOMENCLATURE SIZE INCREASE.....	3-13
3.3.1	<i>Test Description</i> .....	3-13
3.3.1.1	Detailed Description.....	3-13
3.3.1.2	Resource Requirements .....	3-13
3.3.1.2.1	Test Personnel.....	3-13
3.3.1.2.2	Hardware .....	3-13
3.3.1.2.3	Software.....	3-13
3.3.1.2.4	Data .....	3-13
3.3.1.3	Requirements Summary.....	3-14
3.3.2	<i>Pass/Fail Criteria</i> .....	3-14
3.3.3	<i>Procedure</i> .....	3-14
3.4	TEST CASE 3.4 - TEST SSA2 ENUMERATED CLASSES.....	3-20
3.4.1	<i>Test Description</i> .....	3-20
3.4.1.1	Detailed Description.....	3-20

3.4.1.2	Resource Requirements .....	3-20
3.4.1.2.1	Test Personnel.....	3-20
3.4.1.2.2	Hardware .....	3-20
3.4.1.2.3	Software.....	3-20
3.4.1.2.4	Data .....	3-20
3.4.1.3	Requirements Summary.....	3-20
3.4.2	<i>Pass/Fail Criteria</i> .....	3-21
3.4.3	<i>Procedure</i> .....	3-21
	Appendix A Acronyms and Definitions.....	A-1
	Appendix B Requirements Traceability and Test Methods Matrix.....	B-1
	Appendix C Resource Requirements.....	C-1
	Appendix D Standard Test Operating Procedures.....	D-1

## **CSCI INTEGRATION TEST (CIT) PROCEDURES**

### **DATA BANK SHUTTLE AUTOMATED FUNCTION EXECUTIVE (DBSAFE)**

### **CHECKOUT AND LAUNCH CONTROL SYSTEMS (CLCS)**

#### **1. SCOPE**

This document defines the test approach and procedures to be executed for the Thor3 delivery of the Data Bank Shuttle Automated Function Executive (DBSAFE) by CLCS Software Development. Testing will occur at the Kennedy Space Center in the Processing Control Center (PCC) in room 3010K (desktop).

##### **1.1 IDENTIFICATION**

This document is the Checkout and Launch Control System (CLCS) Thor3 Delivery CSCI Integration Test Procedures for DBSAFE Document, 84K06581-001-02, Rev Basic.

##### **1.2 PURPOSE**

The purpose of this document is to define a suite of test procedures that will accurately assess the delivered software to ensure it is functional and meets project commitments for the Thor3 delivery. The CLCS DBSAFE software is software ported from the SDC DBSAFE software; the ported CLCS DBSAFE software is considered the CLCS DBSAFE Baseline. These test procedures will assess any deltas made to the Thor baseline of the ported software.

##### **1.3 CSCI OVERVIEW**

CLCS DBSAFE is a comprehensive Checkout and Launch Control System (CLCS) software capability that provides an interactive user interface supporting the evaluation, incorporation, and historical tracking of engineering changes to the FD Database. DBSAFE for CLCS is ported code baselined from the DBSAFE software developed for the replatform of CCMS Support Software to the Shuttle Data Center (SDC).

The core purpose of the CLCS DBSAFE is to provide the capability to maintain the FD Database. The FD Database is the portion of the CLCS DBSAFE database that contains the information on the measurements, commands, and system parameters needed for CLCS. The attributes of measurements and commands for the orbiters, payloads, ground support equipment, etc., are collected from the various design agencies, processed into a format that is compatible with CLCS, and stored in the FD Database using CLCS DBSAFE software. The data is then available to support the CLCS Application Software Development Environment and Test Build processes.

CLCS DBSAFE also provides the capability to create and maintain TCID build specifications for the FD Directory Build process. CLCS DBSAFE validates and stores

user specifications in the CLCS DBSAFE database. CLCS DBSAFE facilitates the generation of TCID specifications by automating the following functions:

- Assignment and traceability of Vehicle Configuration Names (VCN) and formats to Test Configuration Identifiers (TCID) based on a list of engineering provided by Ground Software Integration (GSI)
- Assignment of projected VCNs and formats to TCIDs based on matching each mission/TCID configuration to the effectivities of engineering changes in the FD Database
- Assignment and traceability of format revisions to each mission/TCID based on the format engineering defined in the Shuttle Data Tape (SDT)
- Support elimination of invalid/duplicate/overlapping addressing in the FD Database, that would otherwise cause errors in a TCID build.

#### **1.4 HARDWARE AND SOFTWARE CONFIGURATIONS**

CLCS DBSAFE software executes primarily on the LPS Software Development Network (LSDN) of Hewlett-Packard UNIX workstations. The CLCS DBSAFE menus and all user interface applications are written using Oracle\*Forms. All reports are initiated from Oracle\*Forms based applications accessed from the CLCS DBSAFE menu. The reports are written in Oracle\*Report and are generated as background processes running on the LSDN workstation.

The CLCS DBSAFE main menu is initiated through the Relational Software Interface (RSI) system resident on the LSDN. RSI is one of several configuration management tools and techniques designed to fulfill the access control and data integrity requirements of CLCS applications. RSI also satisfies the Ad Hoc Query requirements for CLCS DBSAFE by providing read-only access to the CLCS DBSAFE database via either Oracle SQL\*Plus or Oracle Browser.

All CLCS DBSAFE data is stored in an Oracle database located on a Digital UNIX server within the SDC. For testing purposes, it is assumed that all prerequisite CLCS DBSAFE control and legal values data has been loaded into the database. Data supporting the CLCS DBSAFE management services is located in the same physical database as the FD Database; the term "CLCS DBSAFE database" encompasses all. The distinction is important because CLCS DBSAFE supports a much higher degree of configuration management and control over the data within the FD Database. CLCS DBSAFE uses stored database procedures and triggers to satisfy many processing requirements. Communication between the software running on the LSDN and the database in the SDC is handled by Oracle SQL\*Net.

## **1.5 DOCUMENT ORGANIZATION**

This document is divided into three sections and four appendices:

Section 1, Scope, discusses the purpose of the CSCI Integration Test, provides a system overview , and describes software and hardware configurations for the system.

Section 2, Applicable Documents, lists the documents used to create and those supporting this document.

Section 3, Test Case Description, contains a description of the test cases, the pass/fail criteria, and the procedures in detail.

Appendix A, Acronyms and Definitions, contains a listing of acronyms and selected word definitions (for words which may have multiple interpretations)

Appendix B, Requirements Traceability and Test Methods Matrix, contains the requirements verification matrix for the test.

Appendix C, Resource Requirements, contains a list of software, hardware, and personnel requirements necessary for each test.

Appendix D, Standard Operating Test Procedures, contains any specific, standard procedures identified within the test cases.

## 2. APPLICABLE DOCUMENTATION

The following documents, of the revision shown, form a part of this document to the extent specified.

### 2.1 PARENT DOCUMENTS

The documents in this paragraph establish the criteria and technical basis for the existence of this document. The parent documents are:

Parent Document	Document Number	Rev.	Date
CLCS DBSAFE Software Requirements and Design	84K0910-010	B	06/01/1998

Table 2.1: Parent Documents

### 2.2 APPLICABLE DOCUMENTS

Applicable documents are those documents which form a part of this document. These documents, at the revisions listed below, carry the same weight as if they were stated within the body of this document.

Applicable Document	Document Number	Rev.	Date
CLCS DBSAFE Software Requirements and Design	84K00910-010	B	06/01/1998

Table 2.2: Applicable Documents

## 2.3 REFERENCE DOCUMENTS

Reference documents are those documents which, though not a part of this document, serve to clarify the intent and contents of this document.

Reference Document	Document Number	Rev.	Date
CLCS System Level Specification	84K00200	Basic	06/26/1997
CLCS DBSAFE Software Requirements and Design	84K00910-010	B	06/01/1998

Table 2.3: Reference Documents

### **3. TEST CASE DESCRIPTION**

The test cases will validate that all the stated Thor3 requirements have been satisfied. This section describes each test case, the expected results, the pass/fail criteria, and a step by step procedure to execute the test. Appendix B contains the Requirements Traceability and Test Methods Matrix, which maps functional requirements to the test case that verifies those requirements. Test cases are stand alone, and can be executed in any order.

#### **3.1 TEST CASE 3.1 - TEST LOGICAL DESTINATION SYSTEM FDS**

Validate that support for Logical Destination FDs has been implemented.

##### **3.1.1 Test Description**

###### **3.1.1.1 Detailed Description**

This test case will demonstrate that DBSAFE support for Logical Destination FDs has been implemented per the functional requirement stated in section 3.1.1.3. Each step of the test case will demonstrate specific aspects of this support ranging from viewing table structures/data and procedures, to manipulating Logical Destination FDs in the database using the DBSAFE forms.

###### **3.1.1.2 Resource Requirements**

###### **3.1.1.2.1 Test Personnel**

Personnel required include at a minimum a Test Conductor, a QA witness and I&T. Skills required by the test conductor (or designee) include an understanding of the CLCS DBSAFE execution environment and general Oracle knowledge.

###### **3.1.1.2.2 Hardware**

The following Hardware is required:

- Refer to Section 1.4

###### **3.1.1.2.3 Software**

The following Software is required:

- Refer to Section 1.4

###### **3.1.1.2.4 Data**

The following Data is required:

- Refer to Section 1.4

### 3.1.1.3 Requirements Summary

This test case demonstrates that the following functional requirements are met:

Requirement Number	Description
1.2.2.24	<p>Support for Logical Destination System FDs will be provided.</p> <ul style="list-style-type: none"><li>• Add a Gateway column to the clcs_system_fd_cd and dbs_temp_system_fd_info tables.</li><li>• Add a Gateway column to the views for Test Build.</li><li>• Add logical destination types/numbers to the Type table.</li><li>• Modify the dbs 'get-real' procedure to handle the new gateway column.</li><li>• Add Gateway and Source fields to the query screen of the imp_system_fd_def form.</li><li>• A Logical Destination FD must always have a Gateway associated with it.</li><li>• The source field will always have the same value as the FD Type and will be automatically populated.</li><li>• The views supporting the TCS Compiler will not be modified to support Logical Destination FDs.</li></ul>

### 3.1.2 Pass/Fail Criteria

The test case shall fail if any step that is intended to satisfy a functional requirement produces a result that is not equal to the expected result.

### 3.1.3 Procedure

Refer to Procedure 3.1 in Table 3.1 for test procedures.

Procedure 3.1 - Test Logical Destination System FDs					
		Date:	Location:	Start Time:	
Test Setup/Initial Conditions - Refer to Appendix D					
Step	Description	Expected Results	Comments	TC	QA
1.	Via SQL*Plus, a describe will be done on the DBSAFE.SYSTEM_FD_CD table.	The GATEWAY column will now exist.			
2.	Via SQL*Plus, a describe will be done on the DBSAFE.TEMP_SYSTEM_FD_INFO table.	The GATEWAY column will now exist.			
3.	Via SQL*Plus, a describe will be done on the DBSAFE.TBL_SYSTEM_FD_CD_INFO view.	The GATEWAY column will now exist.			
4.	Via SQL*Plus, a select of all Logical Destinations types from the DBSAFE.TYPE_INFO table will be performed. (see comments for actual select statement to be used)	12 Logical Destination FDs will be present: CCP, CGW, DDP, DEU, EIU, GPC, GSE, LDB, PCM, SMEC, SSME, UPLK	Select type, type_name, type_class_name from dbsafe.type_info where type_name like 'LOGICAL DESTINATION%';		
5.	Via Text Editor view the GET_REAL_COMPILER procedure in the DBS_PROCEDURES.SQL script. View the code showing the update of the SYSTEM_FD_CD_INFO table that includes the GATEWAY column.	The update statement for the SYSTEM_FD_CD_INFO table will now include the GATEWAY column.			

6.	From the Implement option of the CLCS DBSAFE Main Menu, select the System FD Definition suboption and select the query button on the form.	The GATEWAY and SOURCE fields will now be displayed on the form.			
7.	From the Implement option of the CLCS DBSAFE Main Menu, select the System FD Definition suboption and attempt to add a Logical Destination System FD to the database without specifying a Gateway.	The add operation will fail due to the fact that the GATEWAY field was not specified. An error message will be displayed and the Gateway field will be highlighted.	CITLD		
8.	From the Implement option of the CLCS DBSAFE Main Menu, select the System FD Definition suboption and add a Logical Destination System FD to the database. Query the FD back using the Query capability of the System FD Definition form.	A SYSTEM_FD_CD_INFO record will exist for the Logical Destination FD and the SOURCE field will have the same value as the TYPE field.	CITLD		
9.	Via SQL*Plus, a describe will be done on the DBSAFE.COMPILED_CD view.	The GATEWAY column will not exist.			

Table 3.1

End Time: \_\_\_\_\_

**Signature Page: Test Case 3.1 - Test Logical Destination System FDs**

---

**Quality Assurance**

---

**Date**

---

**Test Conductor**

---

**Date**

**Comments:**

---

---

---

---

---

---

---

---

---

---

## **3.2 TEST CASE 3.2 - TEST PSEUDO (STRING) FDS**

Validate that support for Pseudo String FD Types has been provided.

### **3.2.1 Test Description**

#### **3.2.1.1 Detailed Description**

This test case will demonstrate that DBSAFE support for Pseudo String FD Types has been provided per the functional requirement stated in section 3.2.1.3. Each step of the test case will demonstrate specific aspects of this support ranging from viewing table structures/data, to manipulating Pseudo String FD Types in the database using the DBSAFE forms.

#### **3.2.1.2 Resource Requirements**

##### **3.2.1.2.1 Test Personnel**

Personnel required include at a minimum a Test Conductor, a QA witness and I&T. Skills required by the test conductor (or designee) include an understanding of the CLCS DBSAFE execution environment and general Oracle knowledge.

##### **3.2.1.2.2 Hardware**

The following Hardware is required:

- Refer to Section 1.4

##### **3.2.1.2.3 Software**

The following Software is required:

- Refer to Section 1.4

##### **3.2.1.2.4 Data**

The following Data is required:

- Refer to Section 1.4

### 3.2.1.3 Requirements Summary

This test case demonstrates that the following functional requirements are met:

Requirement Number	Description
1.2.2.25	<p>Support for a new Pseudo FD Type (String) will be provided.</p> <ul style="list-style-type: none"><li>• Add string as a valid Pseudo FD type (PSTR).</li><li>• Increase size of FD_LENGTH and CONVERTED_LENGTH to Number(4,0).</li><li>• The valid length of a String FD will be 1 – 256 characters specified in bits.</li><li>• A variant record, containing only FD/VCN/DBCODE and TPS File information, will be created for String type Pseudo FDs.</li></ul>

### 3.2.2 Pass/Fail Criteria

The test case shall fail if any step that is intended to satisfy a functional requirement produces a result that is not equal to the expected result.

### 3.2.3 Procedure

Refer to Procedure 3.2 in Table 3.2 for test procedures.

Procedure 3.2 - Test Pseudo String FDs					
		Date:	Location:	Start Time:	
Test Setup/Initial Conditions - Refer to Appendix D					
Step	Description	Expected Results	Comments	TC	QA
1.	Via SQL*Plus, a select of all Types from the DBSAFE.TYPE_INFO table will be performed. (see comments for actual select statement to be used)	A Pseudo String (PSTR) type will be present.	Select type, type_name, type_class_name from dbsafe.type_info		
2.	Via SQL*Plus, a describe will be done on the DBSAFE.FD_LENGTH_INFO table.	The FD_LENGTH and CONVERTED_LENGTH columns will now be sized as Number(4).			
3.	From the Implement option of the CLCS DBSAFE Main Menu, select the Compiler Definition suboption and add a Pseudo String FD to the database. Query the FD back using the Query capability of the Compiler Definition form.	A COMMON_CD_INFO record will exist for the Pseudo String FD.	CITPSTR Demonstrate that the FD Length can have a value between 8 and 2048 (1-256 characters specified in bits.)		
4.	From the Implement option of the CLCS DBSAFE Main Menu, select the Hardware Definition suboption and add Hardware information for the Pseudo String FD. Query the FD back using the Query capability of the Hardware Definition form.	A COMMON_HD_INFO record and an associated HD variant record will exist for the Pseudo FD.	CITPSTR		
5.	Via SQL*Plus, a select of the variant record for the Pseudo String FD Types from the DBSAFE.PSEUDO_HD_INFO table will be performed. (see comments for actual select statement to be used)	There will be a HD variant record with FD_NAME, VCN, DBCODE, and TPS_FILE_KEY columns valued.	Select * from dbsafe.pseudo_hd_info where fd_name = 'CITPSTR'		

Table 3.2

**End Time:** \_\_\_\_\_

**Signature Page: Test Case 3.2 - Test Summary Constraint FDs**

---

**Quality Assurance**

---

**Date**

---

**Test Conductor**

---

**Date**

**Comments:**

---

---

---

---

---

---

---

---

---

---

### **3.3 TEST CASE 3.3 - TEST FD NAME AND NOMENCLATURE SIZE INCREASE**

Validate that support for FD Names with a maximum length of 20 characters and Nomenclatures with a maximum length of 100 characters has been provided.

#### **3.3.1 Test Description**

##### **3.3.1.1 Detailed Description**

This test case will demonstrate that DBSAFE supports larger FD Names and Nomenclatures per the functional requirement stated in section 3.3.1.3. Each step of the test case will demonstrate specific aspects of this support ranging from manipulating FD and Nomenclature data in the database using the DBSAFE forms, to referencing larger FD Names and Nomenclatures in the database using the DBSAFE forms, to running DBSAFE reports.

##### **3.3.1.2 Resource Requirements**

###### **3.3.1.2.1 Test Personnel**

Personnel required include at a minimum a Test Conductor, a QA witness and I&T. Skills required by the test conductor (or designee) include an understanding of the CLCS DBSAFE execution environment and general Oracle knowledge.

###### **3.3.1.2.2 Hardware**

The following Hardware is required:

- Refer to Section 1.4

###### **3.3.1.2.3 Software**

The following Software is required:

- Refer to Section 1.4

###### **3.3.1.2.4 Data**

The following Data is required:

- Refer to Section 1.4

### 3.3.1.3 Requirements Summary

This test case demonstrates that the following functional requirements are met:

Requirement Number	Description
1.2.2.26	<p>Support for FD Names with a maximum length of 20 characters and Nomenclatures with a maximum length of 100 characters will be provided. The increase in the size of FD Name and Nomenclature will allow for the accommodation of Space Station data which may be defined with more than 10 character names and 34 character nomenclatures. In addition, this will permit more descriptive naming of FDs such as those for Pseudos and System Status FDs. The maximum length of Data Set FD Names and the maximum lengths of FD Names and Nomenclatures associated with the SYPREP process will not be changed.</p> <ul style="list-style-type: none"><li>• Modify appropriate CLCS DBSAFE .SQL table create scripts to redefine the FD_NAME and NOMENCLATURE columns. FD_Name will be redefined as VARCHAR2(20); Nomenclature will be redefined as VARCHAR2(100). The affected tables will then be recreated in order to implement modifications.</li><li>• Modify appropriate CLCS DBSAFE Forms to support the new FD_Name and Nomenclature maximum lengths.</li><li>• Modify appropriate CLCS DBSAFE Reports to support the new FD_Name and Nomenclature maximum lengths.</li></ul>

### 3.3.2 Pass/Fail Criteria

The test case shall fail if any step that is intended to satisfy a functional requirement produces a result that is not equal to the expected result.

### 3.3.3 Procedure

Refer to Procedure 3.3 in Table 3.3 for test procedures.

Procedure 3.3 - Test FD Name and Nomenclature Size Increase					
		Date:	Location:	Start Time:	
Test Setup/Initial Conditions - Refer to Appendix D					
Step	Description	Expected Results	Comments	TC	QA
1.	Issue a grep command in the /nfs/hp0247/cssw/ccms_supt/dbsafec/clcs_db directory to find any occurrence of 'fd_name' along with 'varchar2(' (see comments for actual grep command to be used)	A list of all occurrences of the requested string and the associated script filename will be generated.	Grep – i fd_name *.sql   Grep – i 'varchar2('		
2.	Issue a grep command in the /nfs/hp0247/cssw/ccms_supt/dbsafec/dbs directory to find any occurrence of 'fd_name' along with 'varchar2(' (see comments for actual grep command to be used)	A list of all occurrences of the requested string and the associated script filename will be generated.	Grep – i fd_name *.sql   Grep – i 'varchar2('		
3.	Issue a grep command in the /nfs/hp0247/cssw/ccms_supt/dbsafec/clcs_db directory to find any occurrence of 'nomen' along with 'varchar2(' (see comments for actual grep command to be used)	A list of all occurrences of the requested string and the associated script filename will be generated.	Grep – i nomen *.sql   Grep – i 'varchar2('		
4.	Issue a grep command in the /nfs/hp0247/cssw/ccms_supt/dbsafec/dbs directory to find any occurrence of 'nomen' along with 'varchar2(' (see comments for actual grep command to be used)	A list of all occurrences of the requested string and the associated script filename will be generated.	Grep – i nomen *.sql   Grep – i 'varchar2('		

5.	Via SQL*Plus, a describe will be done on the ALL_TAB_COLUMNS and ALL_OBJECT tables.	The tables will exist.	This is done as information for the following steps.		
6.	Via SQL*Plus, a select from the ALL_TAB_COLUMNS table of the table_name, column_name, and data_length fields for column names containing 'FD_NAME' where the owner is 'DBSAFE' will be performed. (see comments for actual select statement to be used)	A list of all columns with names like FD_NAME will be listed along with the length.	Select table_name, column_name, data_length from all_tab_columns where column_name like '%FD_NAME%' and owner = 'DBSAFE' and table_name in (select object_name from all_objects where object_type = 'TABLE' and owner = 'DBSAFE') order by 2,1		
7.	Via SQL*Plus, a select from the ALL_TAB_COLUMNS table of the table_name, column_name, and data_length fields for column names containing 'FD_NAME' where the owner is 'DBSAFE' and the length is not = 20 will be performed. (see comments for actual select statement to be used)	A list of all columns with names like FD_NAME and lengths not = 20 will be listed along with the length.	Select table_name, column_name, data_length from all_tab_columns where column_name like '%FD_NAME%' and owner = 'DBSAFE' and table_name in (select object_name from all_objects where object_type = 'TABLE' and owner = 'DBSAFE') and data_length < > 20 order by 2,1		

8.	Via SQL*Plus, a select from the ALL_TAB_COLUMNS table of the table_name, column_name, and data_length fields for column names containing 'NOMENCLATURE' where the owner is 'DBSAFE' will be performed. (see comments for actual select statement to be used)	A list of all columns with names like NOMENCLATURE will be listed along with the length.	Select table_name, column_name, data_length from all_tab_columns where column_name like '%NOMENCLATURE %' and owner = 'DBSAFE' and table_name in (select object_name from all_objects where object_type = 'TABLE' and owner = 'DBSAFE') order by 2,1		
9.	Via SQL*Plus, a select from the ALL_TAB_COLUMNS table of the table_name, column_name, and data_length fields for column names containing 'NOMENCLATURE' where the owner is 'DBSAFE' and the length is not = 100 will be performed. (see comments for actual select statement to be used)	A list of all columns with names like NOMENCLATURE and lengths not = 20 will be listed along with the length.	Select table_name, column_name, data_length from all_tab_columns where column_name like '%NOMENCLATURE %' and owner = 'DBSAFE' and table_name in (select object_name from all_objects where object_type = 'TABLE' and owner = 'DBSAFE') and data_length < > 100 order by 2,1		

<b>10.</b>	From the Implement option of the CLCS DBSAFE Main Menu, select the Compiler Definition suboption and add a Pseudo Discrete FD to the database. Query the FD back using the Query capability of the Compiler Definition form.	A COMMON_CD_INFO record will exist for the Pseudo Discrete FD.	THISISALONGFDMAXOF20		
<b>11.</b>	From the Implement option of the CLCS DBSAFE Main Menu, select the Hardware Definition suboption and add Hardware information for the Pseudo Discrete FD. Query the FD back using the Query capability of the Hardware Definition form.	A COMMON_HD_INFO record and an associated HD variant record will exist for the Pseudo Discrete FD.	THISISALONGFDMAXOF20		
<b>12.</b>	From the Reports option of the CLCS DBSAFE Main Menu, select the Implement suboption, then select the ONEFD suboption. Request a report of the FD utilized in the above steps.	The report will contain the specified FD. It will present the 20 character FD Name and the 100 character Nomenclature.	THISISALONGFDMAXOF20		
<b>13.</b>	From the Implement option of the CLCS DBSAFE Main Menu, select the Lock/Review (DB) suboption. Lock the TPS Run ID used to perform the previous steps. This will automatically generate a From-To-Report.	The From-To Report generated by the locking process will present the 20 character FD Name and the 100 character Nomenclature.			

Table 3.3

End Time: \_\_\_\_\_

**Signature Page: Test Case 3.3 - Test FD Name and Nomenclature Size Increase**

_____	_____
<b>Quality Assurance</b>	<b>Date</b>

_____	_____
<b>Test Conductor</b>	<b>Date</b>

**Comments:**

---

---

---

---

---

---

---

---

---

---

---

### 3.4 TEST CASE 3.4 - TEST SSA2 ENUMERATED CLASSES

Validate that support for SSA2 Enumerated Classes has been provided.

#### 3.4.1 Test Description

##### 3.4.1.1 Detailed Description

This test case will demonstrate that a SSA2 Enumerated Classes are supported by DBSAFE per the functional requirement stated in section 3.4.1.3. Each step of the test case will demonstrate specific aspects of this support ranging from manipulating SSA2 FD data in the database using the DBSAFE forms, to referencing SSA2 FDs in the database using the DBSAFE forms, to running DBSAFE reports.

##### 3.4.1.2 Resource Requirements

###### 3.4.1.2.1 Test Personnel

Personnel required include at a minimum a Test Conductor, a QA witness and I&T. Skills required by the test conductor (or designee) include an understanding of the CLCS DBSAFE execution environment and general Oracle knowledge.

###### 3.4.1.2.2 Hardware

The following Hardware is required:

- Refer to Section 1.4

###### 3.4.1.2.3 Software

The following Software is required:

- Refer to Section 1.4

###### 3.4.1.2.4 Data

The following Data is required:

- Refer to Section 1.4

##### 3.4.1.3 Requirements Summary

This test case demonstrates that the following functional requirements are met:

Requirement Number	Description
1.2.2.27	<p>Support for SSA2 Enumerated Classes will be provided.</p> <ul style="list-style-type: none"> <li>• The existing System Status Area 2 (SSA2) type will have the capability to have an associated enumerated class.</li> </ul>

### **3.4.2 Pass/Fail Criteria**

The test case shall fail if any step that is intended to satisfy a functional requirement produces a result that is not equal to the expected result.

### **3.4.3 Procedure**

Refer to Procedure 3.4 in Table 3.4 for test procedures.

Procedure 3.4 - Test SSA2 Type FDs Use of Enumerated Classes					
		Date:	Location:	Start Time:	
Test Setup/Initial Conditions - Refer to Appendix D					
Step	Description	Expected Results	Comments	TC	QA
1.	From the DBSAFE Control option of the CLCS DBSAFE Main Menu, select the Enumerated Class suboption and add an Enumerated Class to the database.	An ENUMERATED_CLASS_INFO record and the associated ENUMERATED_ELEMENT_INFO records will exist for the Enumerated Class.	CITENUM		
2.	From the Implement option of the CLCS DBSAFE Main Menu, select the Compiler Definition suboption and add a SSA2 FD to the database. Query the FD back using the Query capability of the Compiler Definition form.	A COMMON_CD_INFO record will exist for the SSA2.	CITSSA2		
3.	From the Implement option of the CLCS DBSAFE Main Menu, select the Hardware Definition suboption and add Hardware information (including reference to an enumerated class) for the SSA2 FD. Query the FD back using the Query capability of the Hardware Definition form.	A COMMON_HD_INFO record and an associated HD variant record will exist for the SSA2 FD.	CITSSA2		

Table 3.4

End Time: \_\_\_\_\_

**Signature Page: Test Case 3.4 - Test SSA2 Type FDs Use of Enumerated Classes**

---

**Quality Assurance**

---

**Date**

---

**Test Conductor**

---

**Date**

**Comments:**

---

---

---

---

---

---

---

---

---

---

## Appendix A    Acronyms and Definitions

Reference Document	Document Number	Rev.	Date
CLCS Acronyms	84K00240	Basic	11/13/1997
CLCS Glossary	84K00250	Basic	11/13/1997

## Appendix B Requirements Traceability and Test Methods Matrix

The following table is intended to show which CLCS Functional Requirement is demonstrated in each CLCS DBSAFE CSCI Integration Test (CIT) and what test method was used in that test case. This table will be updated and baselined with each CIT starting with the Redstone Delivery.

Functional Requirement	Traced SLS Requirement	CI Test	Test Case	Test Method			
				Inspection	Analysis	Demo	Test
1.2.2.24	2.4.2.1, 2.4.2.2	Thor3 CIT	3.1	✓			✓
1.2.2.25	2.4.2.1, 2.4.2.2	Thor3 CIT	3.2				✓
1.2.2.26	2.4.2.1, 2.4.2.2	Thor3 CIT	3.3	✓			✓
1.2.2.27	2.4.2.1, 2.4.2.2	Thor3 CIT	3.4				✓

## Appendix C    Resource Requirements

Refer to Section 1.4

## Appendix D Standard Test Operating Procedures

To gain access to DBSAFE, one must first have a valid userid on the LSDN. Contact the LSDN Help Desk to establish a new account and to obtain any necessary training on how to use a workstation. The Help Desk provides information to help resolve connectivity issues for persons without local access to an LSDN workstation.

Next, obtain an RSI password through the Access Control Data Base Administrator (ADBA) in the USA Quality group. The ADBA must update the RSI authorized access list, establish the Oracle account, and grant authorization to use the DBSAFE\_USER database role. DBSAFE\_USER is a password protected role that provides insert, update, and delete privileges on data and execute privileges on database procedures owned by DBSAFE. When an authorized person initiates DBSAFE, the main menu looks up the password and sets their role to DBSAFE\_USER. This provides the rights needed to modify the DBSAFE database while running DBSAFE applications. This role assignment is terminated when the DBSAFE session ends.

The final steps to obtain access to DBSAFE are managed by the DBSAFE Administrator. This is the person(s) authorized to run the Maintain User Data form to define authorized DBSAFE operators, their rights to execute controlled DBSAFE functions, and their rights to change or delete specific sets of data within the DBSAFE Database. This data is initially checked by the DBSAFE main menu to ensure only authorized people running authorized functions are allowed to modify data. DBSAFE applications access this data as needed to continually enforce privilege constraints defined by the DBSAFE Administrator.

With the following prerequisites met, CLCS DBSAFE can be initiated by typing `rXsi -c f40runmx dbsafec` (with X being d for Development, v for Validation or o for Operations) from a UNIX prompt on the LSDN. The CLCS DBSAFE Main Menu will be displayed, from which the numerous CLCS DBSAFE functions are available.

Also, SQL\*Plus can be initiated by typing `rXsi -c sqlplus` (with X being d for Development, v for Validation or o for Operations).